 doi:10.20944/preprints202209.0103.v1

Article

PDF Malware Detection Based on Optimizable Decision Trees

Qasem Abu Al-Haija[1,*], Ammar Odeh[1] and Hazem Qattous[2]

[1] Department of Computer Science/Cybersecurity, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan; q.abualhaija@psut.edu.jo (Q.A.A-H); a.odeh@psut.edu.jo (A.O)
[2] Department of Software Engineering, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan; h.qattous@psut.edu.jo (H.Q)
* Correspondence: q.abualhaija@psut.edu.jo (Q.A.A-H)

Abstract: Portable Document Format (PDF) files are one of the most universally used file types. This has fascinated hackers to develop methods to use these normally innocent PDF files to create security threats via infection vectors PDF files. This is usually realized by hiding embedded malicious code in the victims' PDF documents to infect their machines. This, of course, results in PDF Malware and requires techniques to identify benign files from malicious files. Research studies indicated that machine-learning methods provide efficient detection techniques against such malware. In this paper, we present a new detection system that can analyze PDF documents in order to identify benign PFD files from malware PFD files. The proposed system makes use of the AdaBoost decision tree with optimal hyperparameters, which is trained and evaluated on a modern-inclusive dataset, viz. Evasive-PDFMal2022. The investigational assessment demonstrates a lightweight-accurate PDF detection system, achieving a 98.84% prediction accuracy with a short prediction interval of 2.174 μSec. To this end, the proposed model outperforms other state-of-the-art models in the same study area. Hence, the proposed system can be effectively utilized to uncover PDF malware at high detection performance and low detection overhead.

Keywords: Portable Document Format (PDF); machine learning; detection; optimizable decision tree; AdaBoost; PDF malware; evasion attacks; cybersecurity

1. Introduction

A piece of harmful code that has the potential to damage a computer or network is referred to as malware. As conventional signature-based malware detection technologies become useless and unworkable, recent years have seen a significant increase in malware. Malware developers and cybercriminals have adopted code obfuscation techniques, which reduce the efficiency of malware defensive mechanisms [1, 2].

Malware classification and identification remain a challenge in this decade. This is largely because advanced malware is more sophisticated and has the cutting-edge ability to remain hidden or change its code or behavior to behave more intelligently. As a result, outdated detection and classification methods are less useful today. As a result, the focus has shifted to machine learning for better malware identification and categorization [3, 4].

Malicious PDF software is one of the common hacking methods [5]. Forensic research is hampered by the difficulty of separating harmful PDFs from large PDF files. Machine learning has advanced to the point where it may now be used to detect malicious PDF documents to assist forensic investigators or shield a system from assault [6]. However, adversarial techniques have been developed against malicious document classifiers. Precision manipulation-based hostile examples that have been carefully crafted could be misclassified. This poses a danger to numerous machine learning-based detectors [7][8]. For particular attacks, various analysis or detection methods have been provided. The threat posed by adversarial attacks has not yet been fully overcome. Figure 1 depicts a PDF document's header, body, cross-reference table (xref), and trailer components [9].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

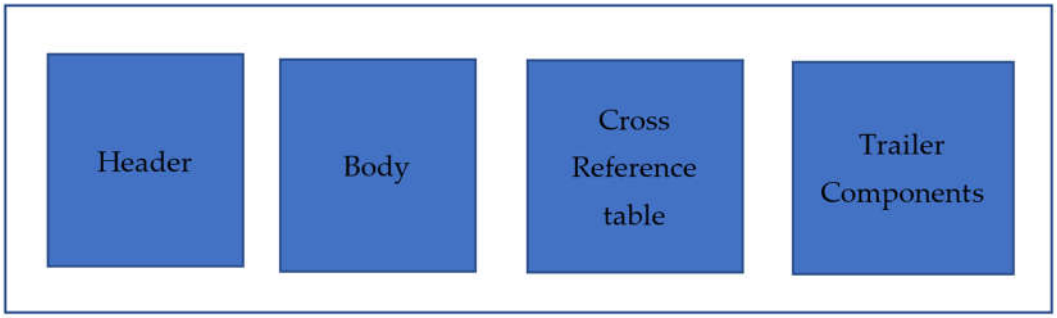

Figure 1. Structure of a PDF file.

The interpreter format version that will be utilized is specified in the header. The PDF's body defines its content and includes text blocks, fonts, pictures, and file-specific metadata. The document's content is contained in a group of PDF elements. These things can fall under one of four categories: Booleans, strings, streams, and numbers [10].

An analyst or analysis tool may use static, dynamic, or hybrid malware analysis techniques (figure 2) [11]. Static analysis techniques examine the sample without running the code and rely on the file attributes, such as the code structure. In analytical methods, dynamism executes the code to observe its behavior, such as the program network operations [12].

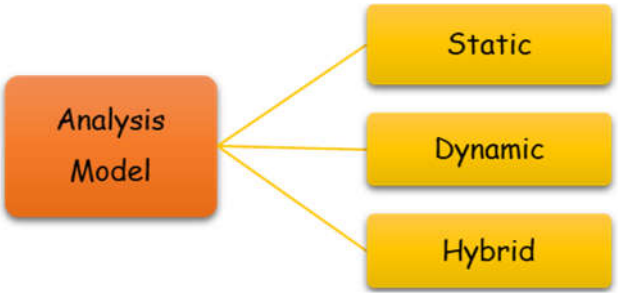

Figure 2. Structure of a PDF file.

Adopting advanced evasion and obfuscation techniques to mask dangerous runtime behavior makes static analysis vulnerable. It is insufficient to undertake static analysis alone in the current security environment. Any serious attacker about their campaign will obfuscate and encrypt their code, typically undetectable by static analysis.

On the other side, dynamic approaches are more resistant to code obfuscation, making them more effective against sophisticated viruses [13]. To avoid harm, dynamic techniques must run the virus in a secure, sandboxed environment. Whether it believes the malware is running in a sandbox or not, an adversary may change the virus behavior to obstruct the malware analysis process [14] [15]. While static analysis is frequently quick, dynamic analysis is typically slow and difficult. Hybrid analysis refers to the combining of the two methodologies. This is more efficient against sophisticated malware than either of the two ways, but it also takes more time and requires a more involved analysis process [16].

In this paper, we present a new detection system that can analyze PDF documents to identify benign PFD files from malware PFD files. The proposed system uses the AdaBoost decision tree with optimal hyperparameters [17], which is trained and evaluated on a modern-inclusive dataset, viz. Evasive-PDFMal2022. The investigational assessment demonstrates a lightweight-accurate PDF detection system, achieving a 98.84% prediction accuracy with a short prediction interval of 2.174 μSec. To this end, the proposed model outperforms other state-of-the-art models in the same study area. Hence, the proposed system can effectively uncover PDF malware at high detection performance and low detection overhead.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

The rest of this paper is organized as follows: Section 2 presents a systematic and inclusive review of the recent related articles in the same field of study. Section 3 provides the modeling architecture for the malware PDF detection system. Section 4 presents and discusses the performance and experimental evaluation results. Lastly, Section5 provides the concluding remarks.

2. literature Review

Deep learning methods, particularly Deep Neural Networks (DNN), have become popular in academic and industrial areas [18]. Their applications can be found in various fields, including malware analysis. On resource-demanding tasks like speech recognition, natural language processing, and picture recognition, DNN performs well. However, it has been demonstrated that machine-learning-based systems categorization is susceptible to hostile settings with cutting-edge evasion attempts [19].

For the identification of malware, supervised machine learning has been frequently used. Several detectors that used this technology were created specifically for PDF files in the past ten years. Choosing whether any unknown PDFs should be classified as harmful or benign is the main objective of machine-learning detectors for malicious document identification. Such systems can work by examining data retrieved from the document's content or structure. Their general process flow is depicted in figure 3, which comprises three main sections [20]: Pre-processing analyzes PDF files and provides access to data essential for detection.

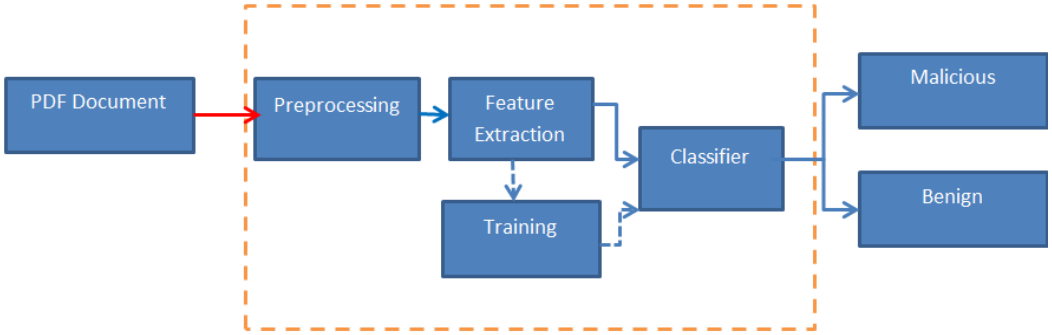

Figure 3. General process flow of machine learning techniques.

The information is transformed into a normalized vector as part of the feature extraction process. To ensure an accurate prediction, the classifier chooses the best learning algorithm for training and modification to acquire improved parameters. Because the quality of the features may have a distinct impact on prediction performance, feature extraction is crucial [21].

An integrated method for malware detection that uses static and dynamic features was introduced by [22]. Combining static and dynamic features has improved identification accuracy compared to using static or dynamic approaches separately. According to the findings, the support vector machine learning method is the most effective at classifying data. However, in addition to improvements in FP and FP rates, the random forest also improved the accuracy [23]. The classification findings show that dynamic analysis is superior to code-based static approaches. In comparison to static approaches, the dynamic method is more accurate. The integrated strategy improves detection accuracy, in line with the study goal.

In [24], the authors proposed a brand-new embedded malware detection system based on statistical anomaly detection techniques. This is the first anomaly-based malware detection method to pinpoint the infection location within an infected file. The suggested Markov n-gram detector outperforms existing detectors in terms of detection rate. Additionally, when used with current COTS AV software, the suggested detector can offer very low false positive rates due to its capacity to locate embedded malware.

 doi:10.20944/preprints202209.0103.v1

A non-signature-based technique that examines the byte-level file content has been proposed by [25]. Such a method offers inherent resistance to typical obfuscation strategies, particularly those that use repacked malware to hide signatures. This study has found that infected and benign files differ fundamentally, even at the byte level. Thirteen unique statistical and information-theoretic features computed on 1-, 2-, 3-, and 4-grams of each file block are used in the proposed approach, which has a rich feature set.

In [26], the authors introduce a framework for machine learning-based robust detection of fraudulent documents. The suggested method is based on elements taken from document structure and metadata. The study demonstrates the suitability of certain document attributes for malware identification and the resilience of these features against new virus strains using real-world datasets. The analysis phase shows that the ensemble classifier Random Forests, which randomly chooses features for each distinct classification tree, produces the highest detection rates.

In this investigation, two main data sources were used. The first is the widely used Contagio data collection [27], which is intended for testing and studying signatures. This source of data sets was chosen because it has many papers classified as malicious and benign, including a sizable proportion from targeted attacks. This source offers a few document sets. The second collection comes from the network monitoring of a sizable university campus. These files were taken out of SMTP and HTTP traffic.

Authors in [28] devised a method to identify a set of features extracted using extant tools and derive a new set of features from improving PDF maldoc detection and extending the lifespan of existing analysis and detection technologies. The derived features are evaluated with a wrapper function that uses three fundamental supervised learning [29] algorithms (Random Forests, C5.0 Decision Trees, and 2-class Support Vector Machines) and a feed-forward deep neural network to determine how important the features are. Finally, a new classifier is built using features of the highest significance, dramatically improving classification performance with less training time. The results were confirmed using sizable datasets from VirusTotal.

The authors of [30] present a brand-new technique for pinpointing an ensemble classifier's data struggles. When enough individual classifier votes conflict during detection, the ensemble classifier prediction is demonstrated to be incorrect. Without the need for extra external ground truth, the suggested technique, ensemble classifier mutual agreement analysis, enables the discovery of numerous types of classifier evasion.

Using PDFrate, a PDF malware detector, the Authors test the proposed strategy and demonstrate that the great majority of predictions can be generated with high ensemble classifier agreement using data from an entire network and our methodology. Nine targeted mimicking situations from two recent research are among the classifier evasion efforts typically assigned an unclear outcome, indicating that the classifier cannot provide a reliable forecast for these data [31]. To demonstrate the approach's broad applicability, the author tested it against the Drebin Android malware detector, where most special attacks were correctly predicted as uncertain. The proposed method can be applied more broadly to reduce the potency of attacks on Support Vector Machines made via Gradient Descent and Kernel Density Estimation. The most crucial element for enabling ensemble classifier diversity-based evasion detection is feature bagging.

The authors in [32] introduce Lux0R, sometimes known as "Lux 0n discriminant References," a unique and portable method for identifying fraudulent JavaScript code. The suggested approach is based on characterizing JavaScript code through references to its API, which includes functions, constants, objects, methods, keywords, and attributes natively recognized by a JavaScript Application Programming Interface (API). The suggested methodology uses machine learning to identify a subset of API references that are indicative of dangerous code and then uses those references to identify JavaScript malware. It has been said that the selection mechanism is "safe by design" against evasion using mimicking assaults. Identifying dangerous JavaScript code in PDF documents is the relevant application domain that the author focuses on in this work.

 doi:10.20944/preprints202209.0103.v1

This technique can obtain outstanding malware detection accuracy even on samples that exploit previously unknown vulnerabilities, i.e., for which there are no instances in training data. Finally, do an experimental evaluation of Lux0R's resistance to mimicking attacks based on feature augmentation.

This work [33] presents a novel approach that combines a feature extractor module closely related to the structure of PDF files with a powerful classifier. This technique has shown to be more efficient than most commercial antivirus programs and other cutting-edge research tools for detecting dangerous PDF files. Furthermore, because of its adaptability, it can be used to enhance the efficiency of an antivirus that is already installed or as a stand-alone program.

It performs significantly better than Wepawet, a potent instrument created by academics. Wepawet has been created to detect various threats, including malicious PDF files, but the developed program is focused on detecting PDF attacks.

It can be further enhanced by assessing the proposed system's resilience to new vulnerabilities and enhancing the parsing procedure. The suggested tool might also be a component of a multi-classifier system, where each classifier focuses on identifying particular dangers. Making our security systems stronger against a wider range of dangers and providing them the ability to anticipate new threats is a challenge for the future as attacker tactics advance.

The authors in [34] discovered the flaws in the existing feature extractors for PDFs by reviewing them and examining how the malicious template was implemented. The authors then created a powerful feature extractor called FEPDF, which can extract features that conventional feature extractors might overlook and capture realistic information about the elements in PDFs. The authors produced many brand-new malicious PDFs as samples to test the current Antivirus engines and feature extractors. The findings demonstrate that several current antivirus engines could not recognize the new harmful PDFs, but FEPDF can extract the crucial elements for enhanced hazardous PDF detection.

This study [35] demonstrates the typical KNN classification algorithm's weaker resistance in adversarial environments by using the gradient descent attack method to alter the malicious samples in the test set to evade detection by the classifier. The authors provide a method in which the created adversarial samples are added to the train set, followed by the usage of the train set to create a new KNN classifier and test their robustness against various attack strengths.

Finally, the tests demonstrate that the robustness of the KNN classifier may be greatly increased without impacting the generalization performance of the KNN classifier by including the adversarial samples produced by gradient descent attack to the train set.

A new data mining-based approach is provided by the authors of [36] introduced for identifying fraudulent PDF files. There are two stages to the proposed algorithm: feature selection and classification. The feature selection step is utilized to choose the ideal amount of features extracted from the PDF file to achieve a high detection rate and a low false positive rate with little computational cost. According to experimental data, a suggested algorithm can achieve a 99.77% detection rate, 99.84% accuracy, and 0.05% false positive rate.

It can perform better by comparing the suggested algorithm against antivirus programs from CalamAV, TrendMicro, MacAfee, and Symantec. The suggested algorithm is based on data mining techniques, which gives it the edge over antivirus software to detect harmful PDF files that have never been seen before. Consequently, the suggested method can better identify advanced persistent threats (APTs).

Using a gradient-descent approach, the naive SVM used by the authors in [37] was easily deceived by us. The authors also devised defenses against this assault by setting a threshold over each considered feature.

This allowed the suggested method to thwart practically all gradient-descent attacks. Next, fewer features were chosen so that features used in the gradient-descent assault could be removed. This reduces the attack's viability even further at the expense of the SVM's precision [38]. The authors also suggested employing adversarial learning to train

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

the SVM using gradient-descent forged PDF files and repeating the procedure to decrease the likelihood that the gradient descent attack will succeed. After only three cycles, the SVM exhibited resistance to attacks using gradient-descent techniques.

Authors in [39] offer in-depth analyses of PDFs' JavaScript content and structure. Then create a rich feature set in JavaScript that includes content features like object names, keywords, and readable strings, as well as the structure and metadata features like file size, version, encoding method, and keywords. It is challenging to create hostile examples when features are diverse because machine learning algorithms are resistant to tiny alterations. To reduce the risk of adversarial assaults, analysts create detection models employing black-box types of models with structure and content properties. Authors create the adversarial attack to verify the suggested model. Additionally, gather wholesome documents with various JavaScript codes for the foundation of the hostile samples.

The PDF files used in this study comprise 9,000 benign and 11,097 malicious document files gathered by the Contagio malware dump between November 2009 and June 2018 [39]. The malware samples are provided via the Contagio malware dump site. From the website, researchers can obtain samples of malware. The samples cover a large amount of time. The authors gathered 115 clean files with JavaScript files separate to develop an adversarial assault for the validation. Authors, except for encrypted files, successfully implanted harmful software into 101 clean files.

In terms of machine learning methods, the authors discovered that while most conventional machine learning algorithms perform adequately for malware detection, they perform worse for adversarial samples, except for the random forest algorithm. Due to this transferability, the random forest algorithm may perform well.

The author in [40] methodically puts forth several guiding concepts to select features to decrease the capacity for escape while retaining high accuracy. These guidelines are followed for extracting features and training a two-stage classifier. The experimental findings demonstrate that our model performs superbly in accuracy, generalization capability, and robustness. It can also differentiate between the vulnerability used in malicious files.

The author introduced a strategy to identify the software that created a PDF file [31] based on coding style: specific patterns that specific PDF producers only produce. Additionally, they looked at the coding practices of 900 PDF files created by 11 distinct PDF producers on three different operating systems. A set of 192 rules that can be used to identify 11 PDF manufacturers has been acquired by the authors. We used 508836 PDF files from scientific preprint sources to test our identification method. The tool used has a 100% accuracy rate for identifying specific producers. Overall, it is still being detected well (74%). To understand how online PDF services operate and detect inconsistencies, utilize the provided tool. Lastly, Table 1 summarizes the important reviewed related research.

3. Proposed Classification System

Portable Document Format (PDF) files are one of the most universally used file types. Like other files such as dot-com files, PNG, and Bitcoin, hackers can find means to use these normally harmless PDF files to create security threats via malicious code PDF files. This results in PDF Malware and requires techniques to identify benign files from malicious files. PDF documents have been seized and exploited as a vector for malicious activities. Abundant PDF readers and software are affected incessantly, for example, CVE-2018-14442, CVE-2017-10994 in Foxit Reader, and CVE-2018-8350 in Microsoft Windows PDF Library [46]. Recent intelligent detection systems are developed via machine/deep learning techniques [47][48] and blockchain/cryptocurrency techniques [49].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

Table 1. Summary of reviewed related research.

Ref.	Model	Datasets	Analysis Model	Advantages	Limitation
[22]	SVM, RF	997 virus files and 490 clean files	Hybrid	The high accuracy rate for static, dynamic, and combined techniques.	Very Small Dataset
[24]	Markov n-gram	37, 000 malware and 1, 800 benign	Static	The Markov n-gram detector offers higher detection and false positive rates than the other embedded malware detection method currently in use.	An evasion test is not available
[25]	(J48) classifier	VX Heavens Virus Collection[42]	Static	The proposed model may identify the malware file's family, such as virus, trojan, etc.	An evasion test is not available
[26]	RF	Contagio [27]	Static	Even though the training set, classification technique, and document features are known, the classifier is resistant to mimicking attacks.	Evasion is much more challenging because classification depends more evenly on many parameters.
[28]	RF, C5.0 DT, and 2-class SVM	Contagio [27] + VirusTool [43]	Static	It gives us a thorough grasp of how these selected features affect classification. And this will improve the training time.	All dataset provided by VirusTotal is benign, and this will make decisions bias
[30]	ensemble classifier (random sampling (bagging))	Contagio [27]	Dynamic	Using Real Data	It does not examine any potential embedded PDF payload
[32]	Heuristic-based	Contagio [27]	Dynamic	More resistant to code obfuscation	Any API extraction mistakes could compromise the accuracy of the detector.
[10]	Bayesian, SVM, J48, and RF	Contagio [27]	Static	multi-classifier system	Not efficient with different types of embedded malicious codes in PDF files
[35]	KNN	Generated Dataset	Static	It drastically lowers false negatives and improves detection accuracy by at least 15%.	An evasion test is not available
[36]	heuristic search, RF, AND DT	Generated Dataset	Static	Identifying advanced persistent threats	It was not tested against evasion techniques and mimicry attacks.
[37]	Naive SVM	Dump [44]	Static	Prevent gradient-descent attacks	Slower than other algorithms
[39]	RF, SVM, and NB	Contagio malware dump between November 2009 and June 2018[44]	Static	Adequately for malware detection	Not detect adversarial samples
[40]	CNN	VirusTotal	Static	Robustness against evasive samples	Can not detect adversarial samples
[41]	Coding style	HAL dataset [45]	Static	Trust generation Process for PDF files	Time-consuming: the complexity depends on the file size.

In this section, we present the proposed detection system used to analyze the PDF files to provide insights into the detection model, which classifies the PFD files into either benign or malware. Figure 4 provides the inclusive architecture for the proposed detection system from the input phase to the output phase.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

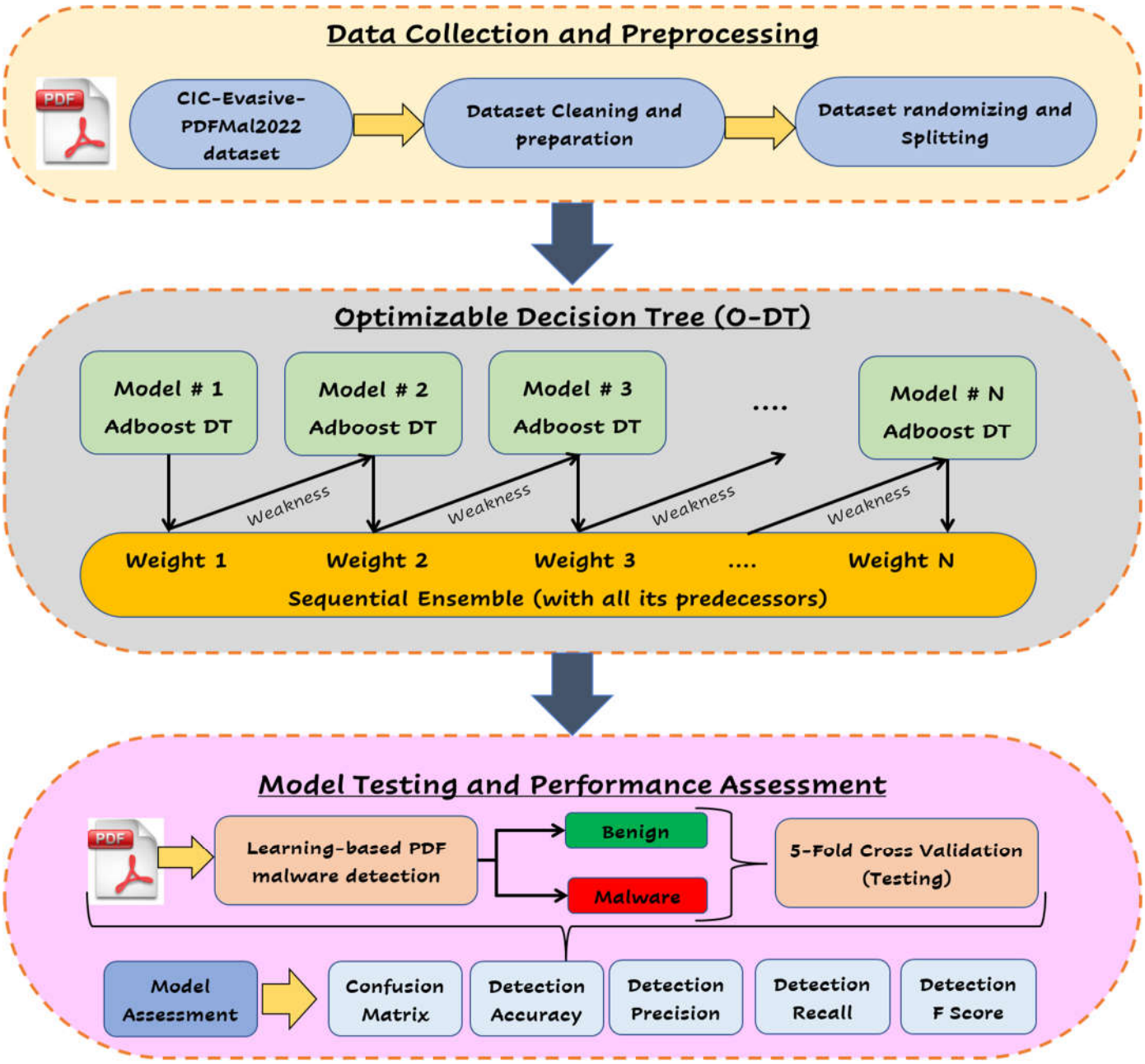

Figure 4. Proposed model architecture.

3.1. Data Collection and Preprocessing

Due to its portability, convenience, and dependability, PDF files are the most commonly used document format for several services and applications. However, this reputation and features of PDFs have attracted Black hackers to harness them in various means. Indeed, a variety of significant PDF features can be exploited by attackers to produce a malicious payload. In this paper, we employ a new and thorough PDF dataset, viz. Evasive-PDFMal2022 comprises 10,025 records distributed as 4468 benign records and 5557 malicious records. Also, Evasive-PDFMal2022 comprises 37 significant static features, including 12 general features and 25 structural features extracted from each PDF file [50]. Examples of features include PDF size, title characters, encryption, metadata size, page number, header, image number, text, object number, font objects, number of embedded files, and the average size of all the embedded media.

The collected data of Evasive-PDFMal2022 is imported via MATLAB 2021 to be processed and prepared for use with supervised learning algorithms. Once imported, the dataset originally available as a comma-separated value (.CSV) file format is hosted by

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

MATLAB as a table of records and features. After that, the data tables undergo a number of data cleaning processes, such as fixing incorrect/incomplete records and removing duplicate/erroneous records. Finally, the data is divided into training (70%), validation (10%), and testing (20%) using k-fold cross-validation (with k =5), as illustrated in Figure 5 below. According to the figure, 20% of the dataset is split out for the final validation of the model. In comparison, 80% of the dataset is used to train and validate the model for several folds. At each fold (say five folds), new random splitting for the 80% is 70% for training and 10% for validation, resulting in 5 different folds of training and validation sets. For each fold, the model is evaluated, and the final overall performance result is the average of the results attained at all folds. To sum up, for our dataset, ~7000 samples are used for training (70%), ~1000 samples are used for validation (10%), and ~2000 samples are used for testing (20%)

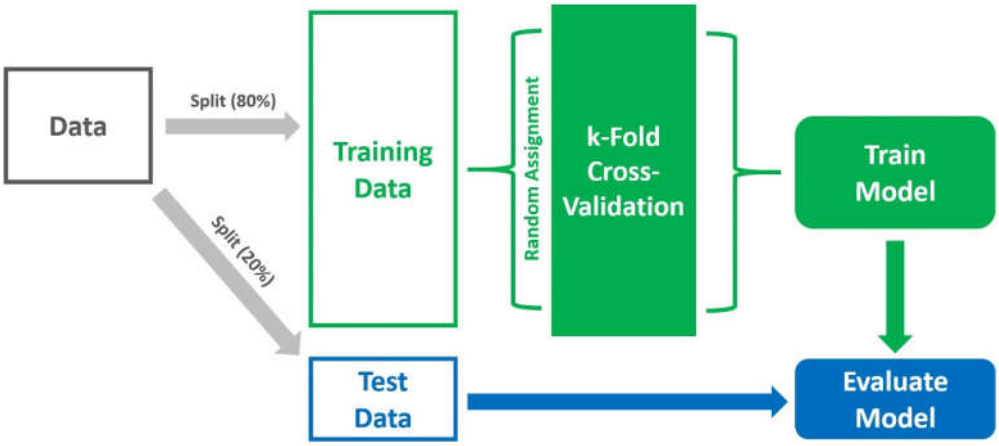

Figure 5. Validation Policy: k-Fold Cross Validation.

3.2. Optimizable Decision Tree (O-DT) Model

A decision tree (DT) algorithm is a non-parametric supervised learning method used to perform classification and regression tasks. DT mainly makes use of a probability tree that facilitates the decision-making of a specific process and can predict the value of a target variable. For example, the need to decide between two project investment ideas can be done through the decision tree. The main idea of DT is to build a model to learn the decision rules inferred from the data features, which can be used later to make decisions and predictions. An optimizable decision tree (O-DT) is a decision tree that makes use of optimal parameters and hyperparameters to build a detection system by trying a predefined search space for different hyperparameters.

We employed the AdaBoost algorithm in this work to build our decision tree model with various hyperparameter options. AdaBoost (ensemble adaptive boosting ML method) is a Boosting approach in which weights are re-allocated to each example, with higher weights allocated to incorrectly classified examples, which helps decrease bias and variance in the learning process. Figure 2 shows how boosting is used in the AdaBoost DT by employing a number of learners expanding sequentially. Apart from the first learner, every successive learner is cultivated from formerly cultivated learners (weak learners are transformed into strong learners). To sum up, table 2 presents the final optimized hyperparameters for developing O-DT.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

Table 2. Optimized Parameters for the development of O-DT.

Factor	Description
Preset	Optimizable Tree
Learning algorithm	AdaBoost Tree
Split criterion	Twoing rule
Surrogate decision splits	Off
Maximum number of splits	6704
Optimizer	Random Search
Iterations	30
Training time limit	False
Feature Selection	All features used in the model, No PCA
Cost function	Minimum Classification Error

3.3. Model Testing and Evaluation

Model testing and evaluation is a crucial process for understanding the performance of a machine learning model and gaining more insights into the model's strengths and weaknesses. In this research, we have tested the model using a 5-fold cross-validation and testing dataset (~2000 samples) and evaluated its performance accordingly. Standard assessment metrics have been used to assess the efficacy of the detection model during the training, validation, and testing phases. Figure 6 summarizes the standard performance assessment indicators utilized in this work.

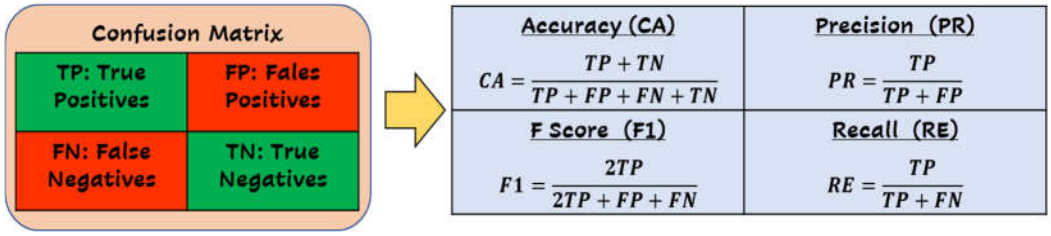

Figure 6. Standard performance assessment indicators.

4. Results and Analysis

This section presents the performance evaluation results for the proposed PDF malware detection system in various indicators. Also, a comparison with state-of-the-art models is conducted. To begin, Figure 7 trace the trajectory of minimum classification error (MCE) during the training iterations of the optimizable decision trees. According to the figure, the initial recorded MCE after the first iteration is 3.4%, recording a maximum classification accuracy of 96.6%. After that, the MCE sharply decreased toward the minimum MCE hyperparameters only after three learning iterations recording an MCE of 1.3% and classification accuracy of 98.7%. Then., the MCE trajectory continued to be slightly decreasing towards the Best-point hyperparameters, where it saturated after iteration 13, recording a 1.16% of MCE with 98.84% of classification accuracy, and it remained constant toward the end of the learning process (30 iterations).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

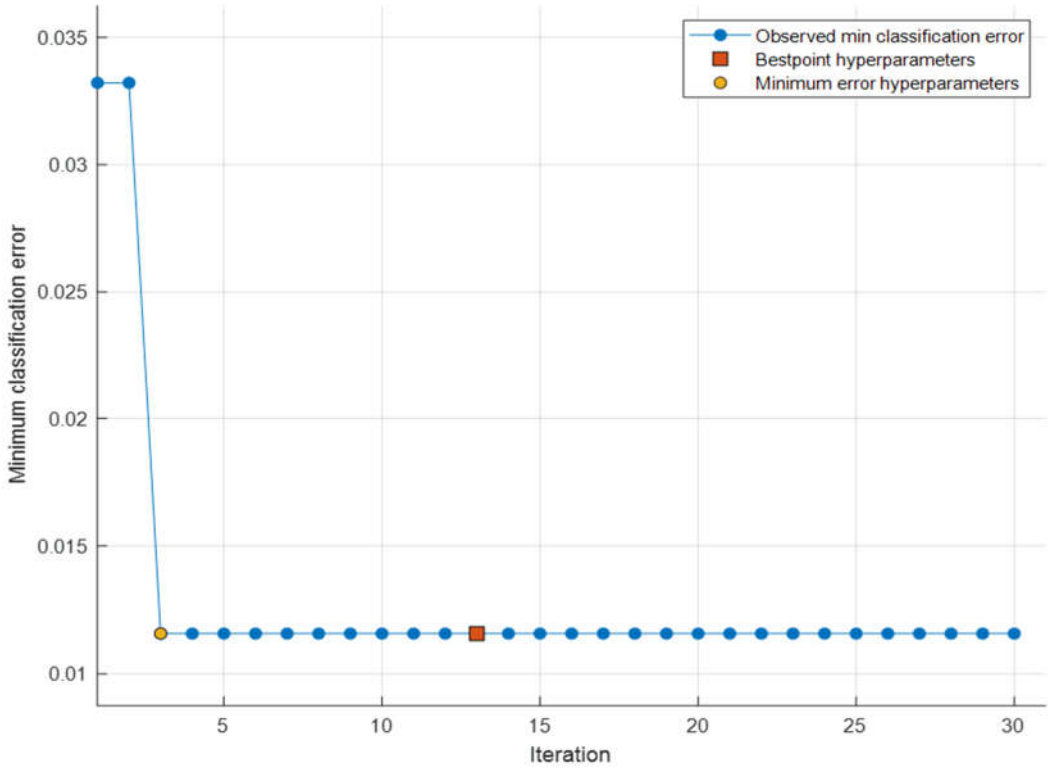

Figure 7. Minimum Classification Error vs. Learning Iterations.

Also, Figure 8 demonstrates the binary confusion matrix results for the proposed PDF malware detection system employing the optimizable decision trees. The presented matrix is composed of blocks: (i) the top left block, which represents the True Positive (TP), expresses the number of samples that are, in reality: Benign samples, and the ML model predicted them as Benign samples. The Number of TP results in this matrix =4412. (ii) the top right block, which represents the False Positive (FP), expresses the number of samples that are, in reality: Benign samples, and the ML model predicted them as Malicious samples. The Number of FP results in this matrix =56. (iii) the bottom left block, which represents the False Negatives (FN), expresses the number of samples that are, in reality: Malicious samples and the ML model predicted them as Benign samples. The Number of FN results in this matrix =60. (iv) the bottom right block, which represents the True Negatives (TN), expresses the number of samples that are, in reality: Malicious samples, and the ML model predicted them as Malicious samples. The Number of TN results in this matrix =5497.

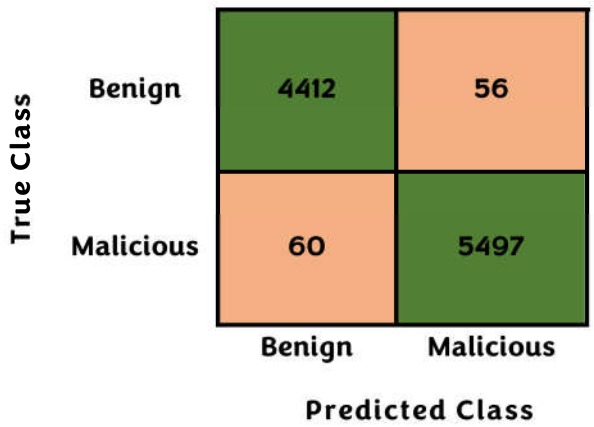

Figure 8. Binary Confusion Matrix Results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0103.v1

Moreover, Table 3 provides a summary of the obtained performance indication results for the proposed PDF malware detection system in terms of various indicators, including Incorrectly Predicted Samples (IPS), Correctly Predicted Samples (CPS), Total Number of Samples (TNS), Prediction speed (PS), Prediction Time (PT), Training time (TT), Prediction Accuracy (CA), Prediction area under the curve (AUC), Prediction Sensitivity/Recall (RE), Prediction Precision (PR), Prediction harmonic average/ F-Score (F1) and Balanced Accuracy (BCA). The attained results exhibit high efficiency and detectability for the proposed system scoring high-performance factors exceeding the 98.80% for the system accuracy, sensitivity, and precision.

Table 3. Summary of experimental evaluation factors for the proposed system.

Factor	Value	Factor	Value
IPS	116 samples	**CA**	98.84%
CPS	9,909 samples	**AUC**	99.00%
TNS	10,025 Samples	**RE**	98.90%
PS	~ 460,000 obs/sec	**PR**	98.80%
PT	2.174 μSec	**F1**	98.85%
TT	11.848 sec	**BCA**	98.95%

Lastly, Table 4 contrasts the performance of our proposed model with several other existing models in the same field of study. The table compares 8 PDF malware detection systems employing diverse learning models, including Zhang. et al. [51] employing a multi-layer perceptron neural network (MLP-NN), Jiang et al. [52] employing a semi-supervised learning algorithm (Semi-SL), Li et al. [53] employing an intelligent tool known as JSUNPACK, Nissim et al. [54] employing support vector machine technique (SVM), Mohammed et al. [55] employing deep ResNet-50 convolutional neural network (ResNet-50 CNN), Nataraj et al. [56] employing random forest classifier (RFC), Lakshmanan et al. [57] employing voting ensemble classifier (VEC) that uses random forest classifier (RFC) and k-nearest neighbor (kNN), and Cohen et al. [58] employing support vector machine technique (SVM). In addition to the learning model factor, we have considered four performance factors to compare with existing models: accuracy, precision, sensitivity, and f-score. Overall, the proposed system is superior in all evaluation factors, with noticeable performance for the other models based SVM technique.

Table 4. Comparison with state-of-the-art models in the same area of study.

Ref.	Model	Accuracy	Precision	Sensitivity	F Score
Zhang. et al. [51] / 2018	MLP-NN	-	-	95.12%	-
Jiang et al. [52] /2021	Semi-SL	94.00%	-	-	-
Li et al [53] / 2017	JSUNPACK	95.11%	97.57%	90.87%	94.10%
Nissim et al. [54]/2014	SVM-Margin	-	-	97.70%	-
Mohammed et al. [55]/2021	ResNet-50 CNN	89.56%	-	-	-
Nataraj et al. [56] / 2020	RFC	96.94%	-	-	-
Lakshmanan et al. [57] / 2020	VEC	95.93%	-	-	-
Cohen et al. [58] / 2019	SVM-Margin	-	-	96.90%	-
Proposed	O-DT	98.84%	98.80%	98.90%	98.80%

5. Conclusions and Remarks

Due to the worldwide trend toward digital transformation and remote work has significantly increased the demand for digital documentation. This increase in the use of digital documents has been obviously accompanied by a counter increase in malware development that can threaten user files and machines. PDF files are among the most commonly used digital files worldwide, which makes them highly vulnerable to a wide range of threats and malicious codes. Such infection vectors (developed by the hackers) hide embedded malicious code in the PDF documents to infect the victims' machines. This results in PDF Malware and requires techniques to identify benign files from malicious files. Therefore, a new intelligent system for PDF Malware detection is proposed, developed, and evaluated in this paper. The proposed system utilized a high-performance machine learning model employing optimizable decision trees with the AdaBoost algorithm. The proposed system was trained and evaluated on a new-inclusive dataset for PDF documents known as Evasive-PDFMal2022. The simulation outcomes showed the superiority of the proposed system in terms of detection accuracy, precision, sensitivity, F-Score, and detection overhead. To this end, the proposed model outperforms other state-of-the-art models in the same study area. The proposed model can be generalized and applied to provide several detection services in various areas [59][60].

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ndibanje, B., et al., Cross-method-based analysis and classification of malicious behavior by API calls extraction. Applied Sciences, 2019. 9(2): p. 239.
2. Abu Al-Haija, Q., Al Badawi, A., & Bojja, G. R. (2022). Boost-Defence for resilient IoT networks: A head-to-toe approach. Expert Systems, e12934. https://doi.org/10.1111/exsy.12934.
3. Ali, M., et al., MALGRA: Machine learning and N-gram malware feature extraction and detection system. Electronics, 2020. 9(11): p. 1777.
4. Faruk, M.J.H., et al. Malware detection, and prevention using artificial intelligence techniques. In 2021 IEEE International Conference on Big Data (Big Data). 2021. IEEE.
5. Ghanei, H., F. Manavi, and A. Hamzeh, A novel method for malware detection based on hardware events using deep neural networks. Journal of Computer Virology and Hacking Techniques, 2021. 17(4): p. 319-331.
6. Atkinson, S., et al., Drone forensics: the impact and challenges, in Digital Forensic Investigation of Internet of Things (IoT) Devices. 2021, Springer. p. 65-124.
7. Liu, C., et al., A novel adversarial example detection method for malicious PDFs using multiple mutated classifiers. Forensic Science International: Digital Investigation, 2021. 38: p. 301124.
8. Al-Haijaa, Q.A. and Ishtaiwia, A., 2021. Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense. International Journal on Advanced Science, Engineering and Information Technology, 11(4), pp.1688-1695.
9. Livathinos, N., et al. Robust PDF document conversion using recurrent neural networks. in Proceedings of the AAAI Conference on Artificial Intelligence. 2021.
10. Wiseman, Y., Efficient embedded images in portable document format. International Journal, 2019. 124: p. 129-38.
11. Ijaz, M., M.H. Durad, and M. Ismail. Static and dynamic malware analysis using machine learning. in 2019 16th International bhurban conference on applied sciences and technology (IBCAST). 2019. IEEE.
12. Chakkaravarthy, S.S., D. Sangeetha, and V. Vaidehi, A survey on malware analysis and mitigation techniques. Computer Science Review, 2019. 32: p. 1-23.
13. Abdelsalam, M., M. Gupta, and S. Mittal. Artificial intelligence assisted malware analysis. in Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems. 2021.
14. Or-Meir, O., et al., Dynamic malware analysis in the modern era—A state of the art survey. ACM Computing Surveys (CSUR), 2019. 52(5): p. 1-48.
15. Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. Appl. Sci. 2022, 12, 5015. https://doi.org/10.3390/app12105015
16. Wang, W., et al., BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. Information Sciences, 2020. 511: p. 284-296.
17. Abu Al-Haija, Q.; Al-Saraireh, J. Asymmetric Identification Model for Human-Robot Contacts via Supervised Learning. Symmetry 2022, 14, 591. https://doi.org/10.3390/sym14030591

18. Al-Haija, Q.A.; Gharaibeh, M.; Odeh, A. Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning. AI 2022, 3, 303-317. https://doi.org/10.3390/ai3020019
19. Yang, L., et al. BODMAS: An open dataset for learning based temporal analysis of PE malware. in 2021 IEEE Security and Privacy Workshops (SPW). 2021. IEEE.
20. Maiorca, D. and B. Biggio, Digital investigation of pdf files: Unveiling traces of embedded malware. IEEE Security & Privacy, 2019. 17(1): p. 63-71.
21. Wu, Y.X., Q.B. Wu, and J.Q. Zhu, Data-driven wind speed forecasting using deep feature extraction and LSTM. IET Renewable Power Generation, 2019. 13(12): p. 2062-2069.
22. Shijo, P. and A. Salim, Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 2015. 46: p. 804-811.
23. Abu Al-Haija Q (2022) Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. Front. Big Data 4:782902. doi: 10.3389/fdata.2021.782902
24. Shafiq, M.Z., S.A. Khayam, and M. Farooq. Embedded malware detection using markov n-grams. in International conference on detection of intrusions and malware, and vulnerability assessment. 2008. Springer.
25. Tabish, S.M., M.Z. Shafiq, and M. Farooq. Malware detection using statistical analysis of byte-level file content. in Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics. 2009.
26. Smutz, C. and A. Stavrou. Malicious PDF detection using metadata and structural features. in Proceedings of the 28th annual computer security applications conference. 2012.
27. Contagio, M.P. http://contagiodump.blogspot.com/2010/08/malicious-documents-archive-for.html. 2011 [cited 2022 02-09-2022].
28. Falah, A., et al., Improving malicious PDF classifier with feature engineering: a data-driven approach. Future Generation Computer Systems, 2021. 115: p. 314-326.
29. Q. A. Al-Haija and K. A. Nasr, "Supervised Regression Study for Electron Microscopy Data," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 2661-2668, doi: 10.1109/BIBM47256.2019.8983101.
30. Smutz, C. and A. Stavrou. When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. in NDSS. 2016.
31. Abu Al-Haija, Q. A Stochastic Estimation Framework for Yearly Evolution of Worldwide Electricity Consumption. Forecasting 2021, 3, 256-266. https://doi.org/10.3390/forecast3020016
32. Corona, I., et al. Lux0r: Detection of malicious pdf-embedded javascript code through discriminant analysis of api references. in Proceedings of the 2014 workshop on artificial intelligent and security workshop. 2014.
33. Maiorca, D., G. Giacinto, and I. Corona. A pattern recognition system for malicious pdf files detection. in International workshop on machine learning and data mining in pattern recognition. 2012. Springer.
34. Li, M., et al. FEPDF: a robust feature extractor for malicious PDF detection. in 2017 IEEE Trustcom/BigDataSE/ICESS. 2017. IEEE.
35. Li, K., et al. Research on KNN algorithm in malicious PDF files classification under adversarial environment. in Proceedings of the 2019 4th International Conference on Big Data and Computing. 2019.
36. Sayed, S.G. and M. Shawkey. Data mining based strategy for detecting malicious PDF files. in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). 2018. IEEE.
37. Cuan, B., et al. Malware detection in pdf files using machine learning. in SECRYPT 2018-15th International Conference on Security and Cryptography. 2018.
38. A. A. Badawi and Q. A. Al-Haija, "Detection of money laundering in bitcoin transactions," 4th Smart Cities Symposium (SCS 2021), 2021, pp. 458-464, doi: 10.1049/icp.2022.0387.
39. Kang, A.R., et al., Malicious PDF detection model against adversarial attack built from benign PDF containing javascript. Applied Sciences, 2019. 9(22): p. 4764.
40. He, K., et al., Detection of Malicious PDF Files Using a Two-Stage Machine Learning Algorithm. Chinese Journal of Electronics, 2020. 29(6): p. 1165-1177.
41. Adhatarao, S. and C. Lauradoux. Robust PDF files forensics using coding style. in IFIP International Conference on ICT Systems Security and Privacy Protection. 2022. Springer.
42. VX Heavens Virus Collection, VX Heavens website, available at http://vx.netlux.org.
43. https://www.virustotal.com/gui/home/upload.
44. dump, C.D.C.M. 2013 [cited 2022; Available from: http://contagiodump.blogspot.com/2013/03/16800-clean-and-11960-malicious-files.html.
45. Available from: https://hal.archives-ouvertes.fr/
46. Priyansh Singh, Shashikala Tapaswi & Sanchit Gupta (2020) Malware Detection in PDF and Office Documents: A survey, Information Security Journal: A Global Perspective, 29:3, 134-153, DOI: 10.1080/19393555.2020.1723747
47. Abu Al-Haija, Q.; Al-Dala'ien, M. ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks. J. Sens. Actuator Netw. 2022, 11, 18. https://doi.org/10.3390/jsan11010018
48. Abu Al-Haija, Q., Al Badawi, A. High-performance intrusion detection system for networked UAVs via deep learning. Neural Comput & Applic 34, 10885–10900 (2022). https://doi.org/10.1007/s00521-022-07015-9

49. Odeh, A.; Keshta, I.; Al-Haija, Q.A. Analysis of Blockchain in the Healthcare Sector: Application and Issues. Symmetry 2022, 14, 1760. https://doi.org/10.3390/sym14091760
50. PDF Dataset. CIC-Evasive-PDFMal2022. Canadian Institute for Cybersecurity (CIC), 2022. Retrieved online: (Access on 1 June. 2022)
51. J. Zhang. MLPdf: An Effective Machine Learning Based Approach for PDF Malware Detection. Cryptography and Security (cs.CR). arXiv:1808.06991 [cs.CR], 2018.
52. Jiang, J. et al. (2021). Detecting Malicious PDF Documents Using Semi-Supervised Machine Learning. In: Peterson, G., Shenoi, S. (eds) Advances in Digital Forensics XVII. Digital forensics 2021. IFIP Advances in Information and Communication Technology, vol 612. Springer, Cham. https://doi.org/10.1007/978-3-030-88381-2_7
53. M. Li, Y. Liu, M. Yu, G. Li, Y. Wang, and C. Liu, "FEPDF: A Robust Feature Extractor for Malicious PDF Detection," 2017 IEEE Trustcom/BigDataSE/ICESS, 2017, pp. 218-224, DOI: 10.1109/Trustcom/BigDataSE/ICESS.2017.240.
54. N. Nissim et al., "ALPD: Active Learning Framework for Enhancing the Detection of Malicious PDF Files," 2014 IEEE Joint Intelligence and Security Informatics Conference, 2014, pp. 91-98, DOI: 10.1109/JISIC.2014.23.
55. T. M. Mohammed, L. Nataraj, S. Chikkagoudar, S. Chandrasekaran and B. Manjunath, "Malware detection using frequency domain-based image visualization and deep learning," Proceedings of the 54th Hawaii International Conference on System Sciences, pp. 7132, 2021.
56. L. Nataraj, B. S. Manjunath, S. Chandrasekaran. Malware classification and detection using audio descriptors, US11244050B2, United States, Jun. 2020.
57. N. Lakshmanan, et al. "OMD: Orthogonal Malware Detection using Audio, Image, and Static Features." MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM). IEEE, 2021.
58. A. Cohen et al., "Sec-Lib: Protecting Scholarly Digital Libraries From Infected Papers Using Active Machine Learning Framework," in IEEE Access, vol. 7, pp. 110050-110073, 2019, DOI: 10.1109/ACCESS.2019.2933197.
59. Q. A. Al-Haija, E. Saleh and M. Alnabhan, "Detecting Port Scan Attacks Using Logistic Regression," 2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT), 2021, pp. 1-5, doi: 10.1109/ISAECT53699.2021.9668562.
60. Abu Al-Haija, Q.; Krichen, M. A Lightweight In-Vehicle Alcohol Detection Using Smart Sensing and Supervised Learning. Computers 2022, 11, 121. https://doi.org/10.3390/computers11080121